Complex Codes Require State of the Art Debugging

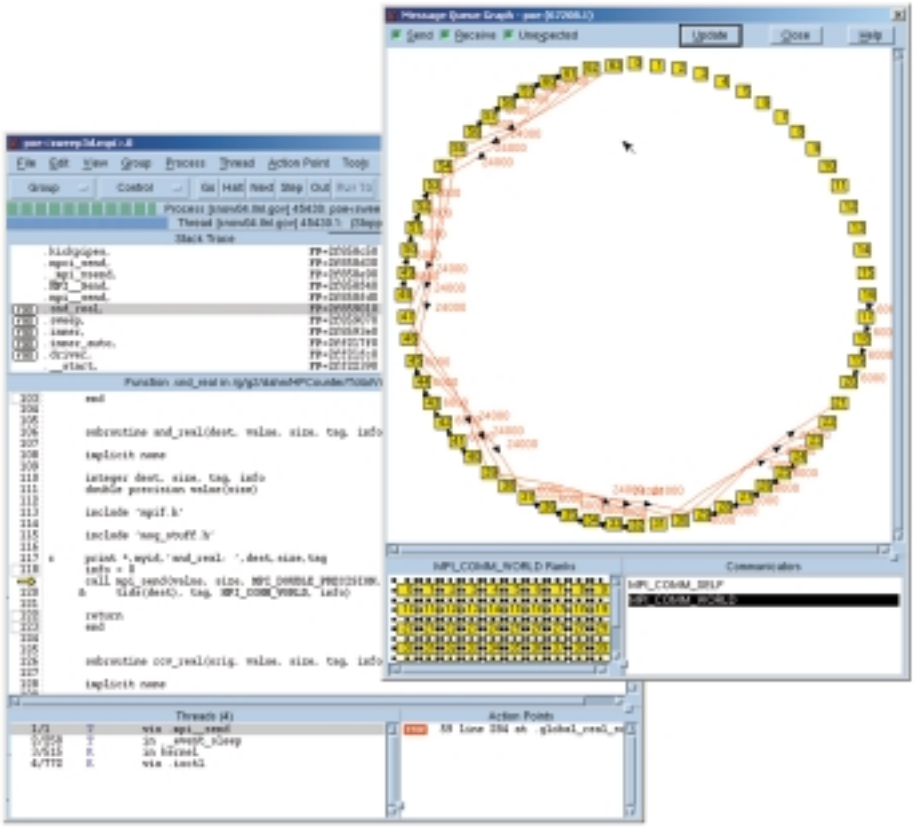

Figure 1 MPI Message Queue Graph

The Department of Energy Accelerated Strategic Computing Initiative (ASCI) program has some of the largest, most complex codes. Along with the increase in raw computing power needed, writing codes that correctly and effectively simulate the aging and operation of a nuclear weapon is core to the ASCI program. Achieving the goals of ASCI happens only by having close partnerships with ISVs and vendors who are committed to the cause.

To meet the demands of debugging the complex software of the ASCI program, ASCI has been partnering with Etnus, through the Ultra Scale Tools Initiative, since 1998. Etnus provides the TotalView debugger to assist in developing the complex and highly scalable applications in this strategic program. Together, Etnus and ASCI develop yearly plans for advancing the capabilities of TotalView. While the features chosen are created for ASCI, Etnus has found that the features they develop for ASCI today are sought after by mainstream markets tomorrow. This too is important for the ASCI program, as another goal is to use commercial products to solve grand challenge applications. In this way, ASCI is helping commercial customers solve their problems tomorrow by paving the path of technology today.

Important Recent Developments

The number and variety of ASCI platforms is continually changing. TotalView's multi-platform flexibility that includes working with native and vendor compilers is key to this program.

Besides sheer scalability in the size of program TotalView can handle, Etnus also adds features that ease the analysis of large codes. TotalView gives the user a global view of the code as it is running. The user can see the current state of all processes via the root window and the call tree graph, and the current data results on all processes, via laminated data displays. TotalView 6.0 will introduce another view into complex programs – that of memory usage.

TotalView Memory Utilization Statistics

Through TotalView's Memory Usage Window, users will see a summary of memory usage for the following: text, data, heap, stack, and the virtual memo-

ry stack. The information will be available on a program and process basis. With this information, you might gain insight into an area of your program that is consuming more memory than you expected.

New Symbol Table

The symbol table is the central nervous system of a debugger. It takes all the information from compilers and executables and builds complex databases of information about your program. To keep TotalView in its current state-of-the-art position, Etnus has rewritten the symbol table system for 6.0. Many new capabilities were made possible by this rewrite, including proper scoping of C++ variables.

Another major new feature made possible by the new symbol table reader is support for executables with non-uniform objects on different processors. If you have processes in a program that were loaded with different versions of libc, for example, typical debuggers will not be able to handle the mismatched addresses within the executable. This is typical of codes in the ASCI program, and this feature allows us to move forward with debugging these codes.

MPI Message Queue Graph

The Message Queue Graph (see figure) visualizes the contents of message queues, so that a user can easily see where process communication may have gone awry. In particularly large codes such as ASCI's MPI codes, this is helpful because the alternative is to take a textual dump of the message queues and devise the order of process communication by hand - a tedious task at best.

TotalView 6.0 expands on its MPI support by offering the ability to attach and detach from subsets of processes in your program. With this feature, you can, without starting your debugging session over, add or delete processes to your debugging session on the fly.

Refined Thread Execution

At times, scalability actually requires concentrating on very small units. Some of ASCI's codes are OpenMP, or mixed MPI and OpenMP. It is very difficult to debug at the worker level (which is actually a thread implementation) of an OpenMP code because it isn't always obvious what the compiler has done to the code during the optimization phase. Etnus has been advancing both OpenMP support and thread support for the last 2 years.

For example, TotalView provides asynchronous thread control, which refines the level at which you can control and debug a threaded program. On most platforms, TotalView lets users debug a single thread in a program, holding all others, or letting them run freely, depending on the circumstance. When ASCI scientists are attempting to find the needle in the haystack, asynchronous thread control can be key.

Research and Planning for the Future

Always pushing the envelop in the state of the art of application development, the ASCI program is ever looking forward to future needs from the tools on which it relies for development of the ASCI codes. Some of the features ASCI is looking to Etnus to provide include: further support for C++ and OpenMP codes, heap allocation debugging, further improving the GUI to more elegantly handle large programs, new visualizations, checkpoint restart, and of course, furthering the performance and scalability for ever-more demanding codes.

For more information on the ASCI PathForward Ultra-Scale Tools Initiative: Scalable Parallel Debugger (Etnus) project, contact kwarren@llnl.gov.

For more information on Etnus, visit www.etnus.com.